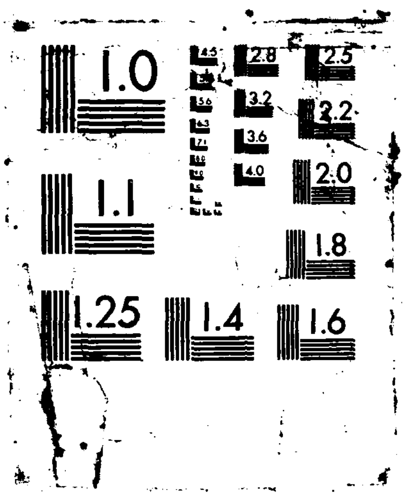
AD-A192 682 SURVEY OF NEURAL NET PARADIGMS FOR SPECIFICATION OF DISCRETE NETWORKS(U) MODAL LOGIC CORP SOLANA BEACH CA M DVORAK 31 JAN 88 1/1

UNCLASSIFIED F/G 12/9 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

②

AD-A192 682

DTIC FILE COPY

SURVEY OF NEURAL NET PARADIGMS FOR

SPECIFICATION OF DISCRETE NETWORKS

Michael Dvorak

Modal Logic Corp.,
P.O. Box 1382
Solana Beach, CA,
92075

31 JANUARY 1988

Interim Report for 1987

Approved for Public Release
Distribution Unlimited

DLAH 4185.8

A192 682

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Modal Logic Corp., — 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code): P.O. Box 1382, Solana Beach, CA, 92075	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification) (U) SURVEY OF NEURAL NET PARADIGMS FOR SPECIFICATION OF DISCRETE NETWORKS

12. PERSONAL AUTHOR(S) Dvorak, Michael

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM 1/1/86 TO 12/31/87	880131	38

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Cognition, Stochastic Processes; Neural Networks; Information theory; Discrete Networks.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Innovative neural network architectures are seen as promising breakthroughs in key problems of interest in image and speech recognition, knowledge base coding and pattern classification. Cost characteristics dictate further research into massively parallel architectures. A survey of some salient characteristics of various paradigms is undertaken in the hope of extracting key underlying organizing principles. There are critiques of continous-type systems, comments on noise, and some cognitive perspectives for discrete networks. There is a brief discussion of memory function. Some stochastic and functional outlines are given in the appendix.

Keywords

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) — 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

(Front Only)

Figure 12

LIST OF FIGURES

I. Introduction

1. General

The term, 'neural network' is generic, encompassing a wide range of physiological and cognitive models, yet cognitive science and neurophysiology are two fundamentally different disciplines involved in what has been termed 'neural' modelling. Both perspectives share some structural characteristics (e.g. thresholding) the differences between the two often center on whether the models should be based on observed physiological properties, or on representations of cognitive processes. Herein we use the term 'neural network' to include any paradigm of a network process that is involved in the recognition or classification of data.

Physiological models involve simulation of known neurophysiological processes [7 et al.], most often 'associative neurons', although from time to time 'motor' and 'sensory' neurons are the focus. Some researchers claim that these simulations can perform tasks such as adaptive pattern recognition and learning.

Although across-the-board performance characteristics have not been determined, there have been some successful demonstrations of these paradigms. These models are typified by polynomial representation.

Cognitive theory presupposes that one can best understand the functions of the mind by seeking to model representations of what are termed 'cognitive processes'[5,6]. The expositions of cognitive models take on a variety of forms, including forms applicable to neural net theory. The field includes such diverse phenomena as intelligence, perception, knowledge, problem-solving, learning and memory, pattern recognition and classification. Consequently, cognitive models lend themselves toward stochastic (random variable) as well as textual presentations. Moreover, the types of patterns and classes that cognitive models use may involve concepts foreign to mainstream research in the field of pattern recognition and classification.

Clearly, both fields are in their infancy, there is some overlap, and hybrids between the two will continue to develop. It is important to gain some kind of grasp of the drawbacks and benefits each perspective presents. Over the past several years there have been a confluence of various disciplines in the field of neural nets. Cognitive science, neural modelling, computer and information science all have contributed to the solution of a particular group of problems. A number of important problems are approached in a manner fundamentally different than that of a typical sequential, (i.e. von Neumann) computer program; these neural architectures and algorithms tend towards parallelism and explicit feedback. Moreover, the neural networks are heuristic; they involve rules-of-thumb, knowledge-seeking behavior, and problems are often attacked from several points simultaneously.

It is believed that 'new' network architectures will aid in the solution of several long standing problems of interest to the AI / cognitive community: handwriting recognition, object recognition, feature identification, and continous time speech, Moreover, these paradigms show promising results in the cognitive field (particularly, in areas such as text understanding, and concept formation).

2. Key Issues in Neural Nets

There are several dilemmas. Of primary concern is that for reasons of cost, it is desirable to have a single network for a variety of applications. Hardware resources must be apportioned according to the nature of the problem. Although it is easily observed that one network will never be optimal for all problems, it is not beyond the realm of possibility that a single network may be able to function effectively over a wide range of problems. It is believed that some degree of speciation is necessary in order to develop those network architectures that will prove most useful. That is, an optimal network will be of a hybrid nature, rather than of a completely homogenous and symmetrical internal design. So on the one hand, there exist a number of different networks in the literature, and it is desirable to survey this material. On the other, there are cost imperatives that indicate consideration of a single model, however internally heterogenous.

This paper contains some basic material as well as some more advanced concepts. Herein we survey various schema and their some corresponding applications with an emphasis on conceptual aspects of imaging. Along the way we will explore and critique various aspects of these paradigms and as well as developing promising avenues for future research. We undertake this research with the intent to contribute to the development of a robust network capable of handling a wide range of problems.

We survey features general to a class of nets; we do not concentrate on one specific application or paradigm, but rather sample from the field those elements which appear promising to more than one models. We are interested in extracting basic underlying concepts, perhaps investigating novel approaches, and then generalizing these concepts to cover a wider range of applications. Each models have their own particular benefits and drawbacks. Our long-term research goals emphasize paradigms, architectures, and general theory that are deemed most promising to various problems of widespread interest, namely, database and knowledge organization, image recognition problems, and continous-time speech. We will not discuss certain problems that are applicable to certain specialized types of network processors, for instance, factory type simulations or communication network models. Nor will we cover certain special applications, such as 3-d imaging, scene segmentation, temporal imaging models, nor phonological analysis of speech.

The cost of problem representation depends on the foundations of the network paradigm and particular characteristics of the domain. This being the case, we feel that rather than attempt to model neural networks on physiological models, there should be detailed investigation of cognitive processes. As has been noted by more than one researcher, 'thought' is something more than just the electro-chemical wirings of the brain. It involves something related to the function of language, that is something that gives those wirings and their signals 'meaning'.

II. Perspectives

1. Some comments

Research in the field of neural networks has resulted in a diversity of models, theories, and notation, an aspect of the research which presents an obstacle as well as an opportunity. Problem domains differ, results and benchmarks that are appropriate across a wide range of systems have not been developed, and combinations or hybrids of networks haven't been explored in depth.

There are a number of different topics that are relevant to the study of 'neural net' theory. Each of these topics are useful tools for analyses of varied neural net paradigms and provide a methodological or structural rooting in terms of research. There are a few central perspectives by which we can grasp fundamental or general properties of neural nets. Any network architecture may be more conducive to one type of analysis over another, but here we present three perspectives that are general threads that run throughout the research, namely: 1) learning automata, 2) 'self-organizing' properties, and 3) pattern recognition and classification. Each of these perspectives have elements in common with the others, however, they serve as reference points by which to view diverse network paradigms.

Learning automata theories envisage a system of learning (or conditioning responses to stimuli), and performance (responses to presented stimuli). In the learning stage, certain knowledge is presented and the system attempts to organize the knowledge into a form in which it can be accurately and readily recalled. The second, performance phase, involves the presentation of a stimulus, ideally, similar to that presented in the learning phase, and by using this stimulus as a kind of key to the organized knowledge, the system generates a response. The learning phase often draws characteristics about the whole data set, (that is the set of presented stimulus-response pairs), whereas, the performance phase involves a one-by-one evaluation of the stimuli. [2,14]. A given system's ability to correctly respond to all presented stimuli depends on the statistical characteristics of the domain. In some instances, there may be noise in the system, manifest as either decision errors, in the discrete case, or inappropriate estimators in the continous case. This state of affairs implies we should view information theoretic concepts such as channel capacity for use in performance measurements.

Self-organization basically seeks to relate a number of objects or nodes in a given network. These relations may be based on adjacency of objects of a certain type (as in imaging applications) or it may seek to characterize a given node by the nature of nodes with which it connects. Self-organizing networks may be represented by varied processes, including instantiation of arcs between nodes, weighting of arcs between nodes and formation of subnetworks. Certain vision based recognition problems are modelled by applying self-organizing principles to adjacency relations. For instance, if two adjacent pixels are of the same color (and not the background color) then we assume the pixels correspond to the same object. This is termed 'blob recognition', or 'object recognition', (although the former is more appropriate for this level of processing). General features such as size of a 'blob' or 'object' may be determined, or contours identified and features extracted from the edge outline [3].

Classification and recognition problems are appropriate for a number of problems; as a body of knowledge there is some overlap with semantic theory. Typical problems involve assigning unique values to those patterns that occur the most frequently, or examining a set of objects one-by-one and then assigning these objects to their appropriate class. Classification may be accomplished by strict rules, or inferred by some process similar to learning. However, not all classification problems fit this form. More intricate assumptions must be made if there is more than one way for a system to validly classify an object. Similarly, classifications based on samples, or classification of objects not previously seen require more sophisticated approaches [8,9 et al].

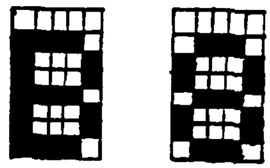
III. Object recognition

1. General

Much of the research into neural models has focused on imaging applications. There are a number of interesting paradigms in this area; here we cover a set of problems encompassed by the rubric 'object recognition'. Object recognition models generally involve identification, or matching of an object with a representation of that type of object in storage. There are two basic approaches: template matching and feature recognition. Template matching involves comparison of an object with a target on a pixel by pixel basis. Feature recognition, involves the use of detection of features and using their presence or absence to distinguish various types of objects. In practice the two perspectives overlap, but the fundamental differences are clear.

2. Template matching

In its simplest form template matching merely compares a given observation with the various templates stored in memory. Comparisons are done on a pixel by pixel basis. The template which most closely matches the observation is chosen as the match for that observation. A perfect match would consist of no errors. If the closest match is not a near match, or if two or more templates are virtual matches with a given instance of input, there may be a problem with noise, that is invalid input----an object that was not intended to be recognized or that was not correctly stored in memory.

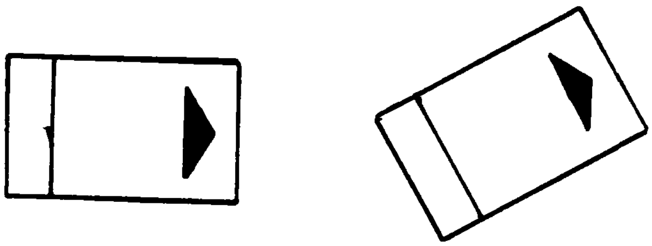


[Fig. 1 Templates of the letter 'B' and the numeral '8']

Typically a Hamming distance model is used, where each pixel is given an equal weight (refer to APPENDIX).

3. Orientation marks

Problems may arise with the template model if there are small rotations of an example from the position as it is registered in the template. In applications such as automated part inspection there may be a significant degree of angular offset, which we refer to as 'orientation'. Now, in some cases, parts have distinguishing features, such as a cut corner or a flange, that serve in and of themselves to determine orientation; in other cases, marks must be correctly placed on the parts in order to ascertain the true orientation. By locating the mark and finding its rotational orientation, the image can be adjusted to the correct position accordingly. This is most useful where the system can't store the various rotations of a large number of different parts, but can recognize various rotations of a single mark applied to different parts.



[Fig. 2 Orientation mark]

4. Feature recognition

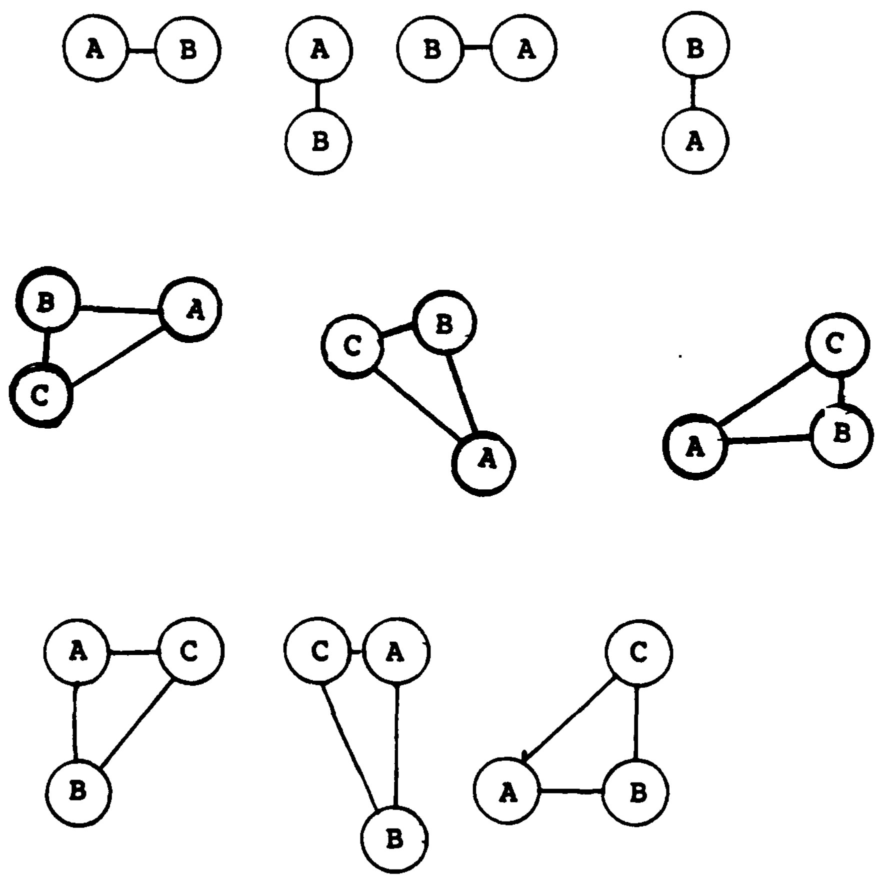
There is the need for a system which does require an explicit orientation mark, yet can orient and identify the various patterns without regard to rotation. For instance, a system that can view and recognize parts on an assembly line, or matching fingerprint with a fingerprint file. Feature recognition can be viewed as a more generic form of recognition than template matching. Rather than matching on a one-to-one basis as occurs in template matching algorithms, feature matching involves identification of formal attributes, such as curves, loops, and straight lines, or, more abstract relations such as correspondences between various attributes previously identified.

This descriptive procedure is more appropriate for many problems, since it allows for a wider range of phenomena to be recognized, and allows for instances that have not been previously seen to be categorized correctly. Ideally, features are drawn from an example, and then these features are matched with a stored description of a class. This stored description may be learned from a sample data set, or in what may amount to the same thing, derived from statistical measurements of the sample. Feature recognition proves useful in recognizing objects that are members of a given type, but where there are case-to-case variations. As an example, recognizing a given printed letter of the alphabet in several different fonts. The key questions revolve around what features are recognized, and how this recognition is accomplished.

Features must be recognized in all images in which they exist. Spatial orientation of the sample should have no effect on identification, nor should the juxtaposition of other objects. Moreover, the system should be able to accurately identify these features or their absence in the presence of some noise, and should be able to identify those cases where excessive noise makes accurate identification impossible.

As a brief example we will outline one simple image identification system involving feature recognition and their topological relations. We assume, specific features of each image are determined by some method (irrespective of rotation) and then these features and certain topological relations, discussed below, are coded and indexed into what amounts to a database. The examples below are somewhat simplified but serve to illustrate the point that even a simple system structured in this manner is capable of distinguishing among a number of similar images which vary only in the arrangement of features. (See APPENDIX for details)

Consider the changes of a two-dimensional front-facing view of a flat object. Intuitively, we know that adjacency is preserved; that is two objects, or features that are adjacent remain adjacent when the observation is rotated. By using a distance relation, (also invariant under rotation) we can gain even more precise matching.



[FIG.3 TOP: Adjacency relations over rotation MIDDLE: four rotations of a given object BOTTOM: three similar but different objects]

5. Staged Feature Recognition

For enhanced utilization of hardware resources, let us consider the structure of a basic staged feature recognition model. In its most basic form, we envision a system embodying two levels of pattern detection, one coarse and the other involving detail, where the output of the 'coarse' recognizer may indicates whether or not further processing is required by the second stage, the 'detail' recognizer before arriving at a conclusion. Typically, we will assume the coarse recognizer provides for much more rapid processing than the detailed recognizer, and that the system is such that there will be no computational overload. We also suppose that the matching task requires a simple yes/no response, but that the matching is of sufficient complexity that we require feature recognition rather than template matching. We will also assume that 'close doesn't count' in the final tally; that we are looking for exact matches. In this paper, we leave out the details of how the features are recognized. We are mainly concerned with the fact that the outputs are determined in some way.

TABLE 1.

	Coarse	Detailed	
Status	Possible Outputs	Input from Coarse	Possible Outputs
x==y	1,2	1	4
x:=y	2	2	4,5
x!=y	3,2	3	5

== identical := approximately identical != not identical

where 1,2,3 are the outputs from the coarse processor

4 indicates a match

5 indicates no match

Now here we see that states 1 and 3 are unambigous, so they need no further processing. (state 1 always maps to 4 and state 3 always maps to 5) And we see that the detailed processor ends up disambiguating ambgious symbols, akin to some fuzzy types of processing. [19,20]

Now by appropriately coding of features and their relations, some kind of statistical classification can often be arrived at that can reduce the search time of the detailed comparisons. The keys are how to accomplish this coding, and the subsequent recognition process.

The target probabilities for a given coarse search:

Table 2.

Searched-by-fine	Having char.	Probability (target)
0	0	don't care
0	1	0
1	0	low
1	1	don't care

The probabilities are of course influenced by the algorithm and the probability distribution of the domain.

Various other similar methods may be used to accomplish the recognition goals. And, of course, hashing and other indexing schemes have been developed that aid in rapid matching of features once the feature database has been generated. However, the simple model above embodies several conceptual issues that may be bases for extensions, three in particular are worthy of brief discussion.

We can envision multi-stage recognition systems such that successive stages resolve indecision. That is each stage's output represents a subset of the previous stage's output. The key performance parameters here involve the cost of processing at each stage and the efficiency of each stage of the process as measured by the amount of ambiguity removed. There may also be cases where a certain bayesian component must be identified.

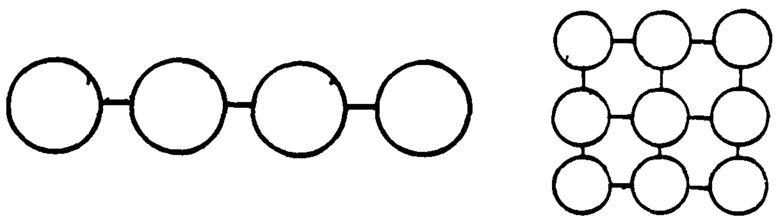
Similarly we can also envision systems that use data from previous levels in order to fine tune their search patterns. That is a system where the output from one level represents not only information about the image, but also information as to relevant search patterns that will contribute to the determination of the final set of features.

There is also the general issue of prediction and control of incoming data. In a manner, similar to information and control models we can analyze these systems by their ability given valid input data, to match or tend toward certain target probabilities.

These three concepts are of interest to our research, not only because of their applications to imaging problems, but because of their use as a framework for more general types of characteristic processing.

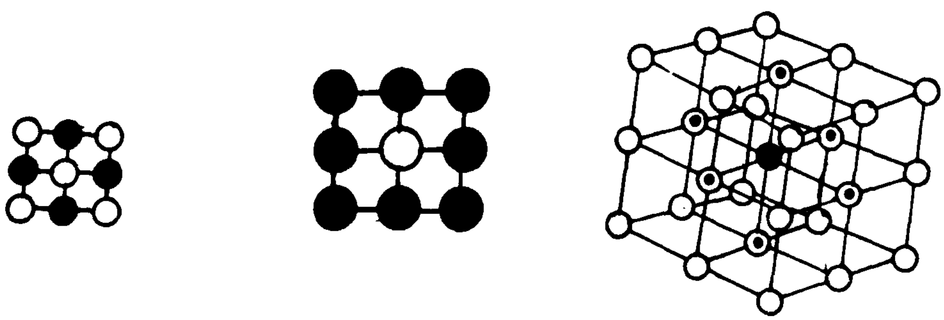
IV. Self-organization facets in imaging

Self-organizing network theory provides some techniques for rapid determination of topological relations. In the form we consider here, self-organization refers to the use of known node and arc relations in order to assign nodes to classes. We should distinguish between self-organization of networks in general, lattice-like nets, and cartesian networks. In the visual case we generally are interested only in the relations between the nearest neighbor nodes in a 2-dimensional space. This self- organization can also be viewed in some cases as a categorization problem, but for imaging applications it is often more convenient to expose the problem initially as a 2 dimensional array of nodes.



[Fig. 4 Locality: One and Two Dimensions]

the most common algorithms for 2-dimensions are a 8-nearest neighbor (a.k.a octree) and 4-nearest neighbor (a.k.a cross); for 3 dimensions we would use 6-nearest neighbor or 26-nearest neigbhor:



[FIG. 5 4 nearest(2-D), 8 nearest (2-D) and 26 nearest (3-D) neighbors respectively.]

These types of nearest-neighbor algorithms are of use in determining size and shape of objects, edge/boundary detection, or determining 'at a glance' whether or not two points correspond to the same object. It has important industrial uses in automated part inspection.

There has been considerable work in theoretical computer science on various aspects of graph theory; some of this work is applicable to theories of self-organizing networks, although imaging and speech applications typically involve the Cartesian restrictions on topology similar to those outlined above.

Although neural nets have some ability to distinguish between objects [12], it is not clear how this self-organizing ability takes place. Moreover, there may be significant channel capacity restrictions, that is, beyond some point, the nets cannot distinguish among a wide variety of images.

V. Network organization

1. General

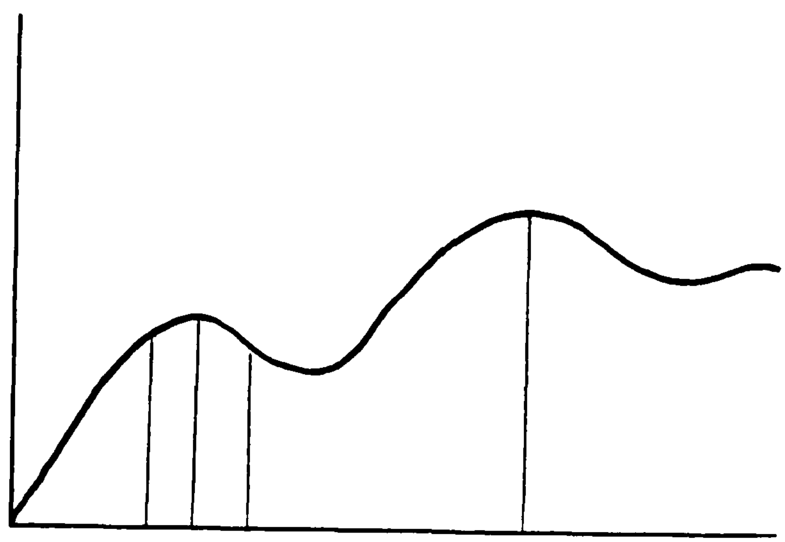
Different paradigms may embody fundamentally different assumptions about network organization. Most models involve a feedback based design; this aspect is more explicit in some models, more covert or formal in others [1,4,8,9,12 et al]. System values are arrived at by explicit numerical processing; we shall call these systems, 'continous-type' systems. Often outputs are not determined by a predetermined sequence of operations, but on state characteristics of the system in general; the 'settling conditions'. These models bear striking resemblance to information and control models of industrial processes.

Systems of polynomial equations often occur in the literature on neural nets, typically taking one of four forms:

- functions constructed by the network as a response to a conditioning sequence and possibly input stimuli; these functions estimate the output value(s) or stand for internal decision or influence processes.
- functions which represent a topological description of the network in whole or in part
- functions which generate arguments for those estimator functions, for instance, functions whose extreme values represent coefficients for the decision functions.
- empirical or estimated performance graph of a neural net over some parameter(s)

The architectures of continous-type systems are required to have at hand a great deal of floating point type hardware; this type of 'number crunching' is quite natural as a model of a physical process; however, in the present context it may have some flaws. The primary critique of this continous-type approach rests on three main issues: 1. that of local versus global min/max and 2. that of absolute performance, or performance in comparison with other nets, given a paradigm performing at optimal levels. 3. that of computational resource utilization

When determining a min or max of a given polynomial we must be assured that we are considering the global min or max and not some intermediate local value:



[FIG. 6 Global vs. local min and max]

As can be seen from the figure, using a too narrow an sampling interval of possible maxima or minima can lead to a local max or min being chosen over a global one. On the other hand, a large number of points cannot be sampled due to combinatoric explosion, and of course, these problems become magnified as the dimensionality is increased.

A second serious issue with continous-type systems is that given a system has reached an optimal state, there is no guarantee that the output will be the 'correct' one. Another way of putting this, is that the scope of optimality of a particular algorithm/architecture is generally narrowly restricted. These 'ideal cases' cannot always be achieved and the 'optimality' generally refers to a performance measurement over a wide domain; it does not guarantee optimality or even correctness of a given output instance.

Moreover, there must be considerations given to the allocation of computational resources. We must balance out computational requirements with that of resource cost and actual performance. We must be sure not to engage in 'computational overkill', but partition the resources according to where they are needed the most.

There are other qualitative issues as well. We should also look at how a given system description performs over various types of input. A system may not be able to distinguish between similar types of objects if it has not seen these instances before; it may be slow to learn the subtle characteristics that distinguish one type of object from another and only have latched onto gross differentiating aspects of these characteristics that have been gleaned from previous instances. In a related vein, neural net models often cannot take into account more than 2nd order effects (i.e. 2 input paths at once), so that certain multivariate events cannot be effectively modelled without more extensions to the architecture.

2. Discrete processes

The 'continous-type system' approach appears to be ill-suited for representing discrete processes (since there exist discontinuities at these points), or discrete state representations rather than solely numerical values. Moreover, this approach does little to distinguish neural net models from vector or array processors.

Remember we are not always assuming that the networks have continous inputs and clearly with a finite number of samples there will be discontinuities between values, and these discontinuities have to be evaluated in terms of overall theoretical framework, particularly if there are analyticity and smoothness assumptions.

3. Memory Mechanisms

The concept of memory mechanisms are of vital interest to cognitive researchers. In its basic form it is concerned with representations of storage and retrieval of information. More complex theories involve information processing or classifciatory procedures as a prelude or adjunct to storage and retreival[10,11,15].

Memory issues are relevant to many of the paradigms we consider. In most of the continous-type models, there is no explicit memory mechanism for the storage of intermediate results. Now, clearly, many of these models have an implicit memory component, in the same sense that expectation is a 'memory' of values of a random variable and their corresponding probabilities, or variance is a 'memory' of concentration of a random variable about its mean. (This would apply to weighted means as well.)

Now the efficiency of this approach depends to a great extent on the internal architecture of the neural net, as well as the 'encoding paradigm'; the relations that determine how particular input-output pairs are encoded into memory.

Traditional models of human memory have pointed out the existence of different types of memory (long-term, short-term, episodic, procedural, textual, visual and so on) [11,15,17], and the encoding, contextual, and classification systems used. Some of the simpler models involve basically a read/write store. This can take several forms, but the one we shall consider here: a model of memory taking the form of an association list such that for a given 'name' a particular 'value' is associated with that name. This 'value' can be read or written by some memory management device. Other paradigms involve content-addressable features.

4. Discrete Networks

This leads us to consider discrete-type cognitive models as a structural basis for neural nets. There are several reasons for this. Many of the phenomena we are considering are discrete: decisions, events, existence of specific attributes, and set membership.Discrete representation would also be indicated where

Many of the phenomena we are considering are discrete: decisions, events, existence of specific attributes, and set membership. Discrete representation would also be indicated where state space representation is required and there is also the motivation to allocate hardware resources; any reasonably priced system would have a limited availability of parallel floating point capacity, so therefore we would not wish to squander these types of computational resources when simpler structures would suffice. As an added bonus input and output alphabets of the systems are usually discrete, and with the possible exception of some of the newer optical processors the machine languages are binary.

When we consider discrete networks, we notice that there are a number of possible benefits. For one, we are not restricted to analytic, differentiable, Riemann integrable functions in modelling of internal processes, and we are now free to investigate synthetic properties of signals, and various types of memory simulation.

Conclusion

There appears to be an inherent flexibility in discrete type representations. They seem to correspond with what cognitive researchers mean when they refer to 'signals'. And, importantly for implementation, discrete models are useful from a numerical as well as a taxonomic point of view, as well as providing more flexibility than models based mainly on floating-point hardware. A few brief comments regarding these models are developed in the appendix.

References

1. Cambier, J.L., Barth, S., Reid, W.J., Barrett,S., "Advanced Pattern Recognition", F30602-80-C-0319. ADA 132 339

2. Elman, J.L. & Zipser, D., "Learning the Hidden Structure of Speech", ICS 8701

3. Grossberg, S. & Mingolla, E., "Neural Dynamics of perceptual grouping: textures, boundaries, and emergent segmentation", ADA 165 648., DAAG-29-85-K-0095.

4. Hecht-Nielsen, R. "Performance Limits of Optical, Electro-Optical, and Electronic Neurocomputers", TRW/RC/AIC, U.S. Army Research Office, DAAG-29-85-C-0025, (DTIC ADA 180 442)

5. Lakoff, George., "Categories and Cognitive Models", BCSR, No. 2, Nov., 1982.

6. Lakoff, George, "Cognitive Semantics", BCSR No. 36

7. Levy, William B., "Direct Assessment of Synaptic Modification Rules", AFOSR-83-0236 ADA 179 548

8. Lippmann, R.P., Gold, B., Malpass, M.L., " A comparison of Hamming and Hopfield Nets for Pattern Classification", group 24, Technical Report 769, MIT Lincoln Labs, May 1987, ADA 182 255.

9. Longstaff, I.D., & Cross, J.F., " A pattern recognition approach to understanding the multi-layer perceptron", Royal Signals and Radar Establishment Memorandum 3935, ADA 174-308.

10. McClelland, J.L., and Rumelhart, D., "Distributed Memory and Representation of General and Specific Information", from Jl. of Exp. Psych. 1985 v. 114, no. 2, 159-88

11. Minsky, Marvin, "K-Lines, A theory of memory", Cognitive Science, 4, 117-33 (1980).

12. Myers, M., Kucsewski, R., Crawford, W., "Application of New Artificial Neural Systems Information Processing Principles to Pattern Classification", TRW/RC/AIC, U.S. Army Research Office, DAAG-29-85-C-0025, (DTIC ADA 180 442)

13. Norman, Donald, & Rumelhart, David, "Human Cognition and Performance", ONR N-00014-79-C-0323 ; NR 667-437, ADA 157 665

14. Rumelhart, D., Hinton, G., and Williams, "Learning Internal Representations by Error Propogation", ICS/UCSD 8506, 9/85, Cognitive Science, 9, 75-112, 1985.

15. Rumelhart, D.E. and Ortony, Andrew, "The representation of knowledge in memory", in Anderson, R.C., Spiro, et al., "Schooling and the acquisition of knowledge", Erlbaum, Hillsdale, NJ, 1977

16. Rumelhart D., & Zipser, D., "Feature Discovery by Competitive Learning", Cognitive Science, 9, 75-112, 1985

17. Schank, Roger, "Language and Memory", Cognitive Science, 4, 243-84 (1980).

18. Williams, R., "Unit Activation Rules for Cognitive Network Models", ICS, 8303.

19. Zadeh, L., "Syllogistic Reasoning in Fuzzy Logic and Its application to usuality and reasoning with dispositions", BCSR No. 34, Dec. 85.

20. Zadeh, L., "A simple view of the Dempster-Shafer Theory of Evidence and its implications for the rule of combination", BCSR No. 33, Nov. 1985.

Bibliography

Berge, Claude, "Principles of Combinatorics", Academic Press, 1971

Fischer, Roland, "Deconstructing Reality", Jl. Biological and Social Structures, 1985 / 8, 105-8

Freund, J., Mathematical Statistics, Englewood Cliffs, NJ, Prentice-Hall, 1982.

Fu, King Sun, "Applications of Pattern Recognition", CRC Press 1982

Groner, Rudolf, Groner, M., and Bischof, W., "Methods of Heuristics", 1983 Erlbaum Associates,

Hammond, K.R., "The Integration of Research in Judgement and Decision Theory", U of Col./IBS/CRJP No. 226, ADA 088 471

Hoel, Port and Stone, "Introduction to Probability Theory", Houghton-Mifflin, 1971

Melton, A.W., and Martin, E. (eds.), Coding Processes in Human Memory

Norman, Donald, "Twelve Issues for Cognitive Science", Cognitive Science, 4, 1-32, 1980

Ogden, C.K., and Richards, I.A., "The Meaning of Meaning", Harcourt-Brace, 1927.

Scholz, Roland W., "Cognitive Strategies in Stochastic Thinking", D. Reidel, 1982(?)

Searle, John, "Expression and Meaning", Cambridge U. Press, 1979

Talmy, Leonard, "The Relation of Grammar to Cognition", Berkeley Cognititve Science Report 45, Nov. 86.

Van Ryzin (ed.) Classification and Clustering, Academic Press, 1977

Whorf, Benjamin, "Language, Thought and Reality", 1956, MIT Press

Wilensky, R. "Some Problems and Proposals for Knowledge Representation", UCB/CSD 86/294, EECS.

APPENDIX

1. Some quantitative representations of fuzzy functions

Here we briefly consider some aspects of quantified fuzzy estimators. These structures give us a method of embodying some imprecision while maintaining certain quantitative relations (see order relations below)

The basic model we shall consider here is such that the range of the estimator Y represents a set of n possible contigous values of the underlying variable X, each having an equal probability.

A single value of a fuzzy function maps to a set of uniformly distributed possible target space values:

$$P(X=i|Y=k) = 0, \quad 0 \le i < k$$
$$P(X=i|Y=k) = 1/m, \quad k \le i \le k+m$$
$$P(X=i|Y=k) = 0, \quad k+m < i \le n \qquad \text{(A1)}$$

In a figurative sense, we can view this type of function as a hybrid type of uniform discrete density.

Given n objects $X_{1...n}$ and with q-samples taken contigously (i.e. $(X_{1...q})...(X_{p...n})$)
$p = n-q+1$, where p the number of q-samples of n objects so we see that values

π	$N(X):P(X)=\pi$
$1/\lambda$	2
$2/\lambda$	2
. . .	. . .
m/λ	ξ

with m/λ the probability of the ξ maximum likelihood estimators of the target function.

From this model we get the following algebraic relations:

$m = \min(p,q)$,

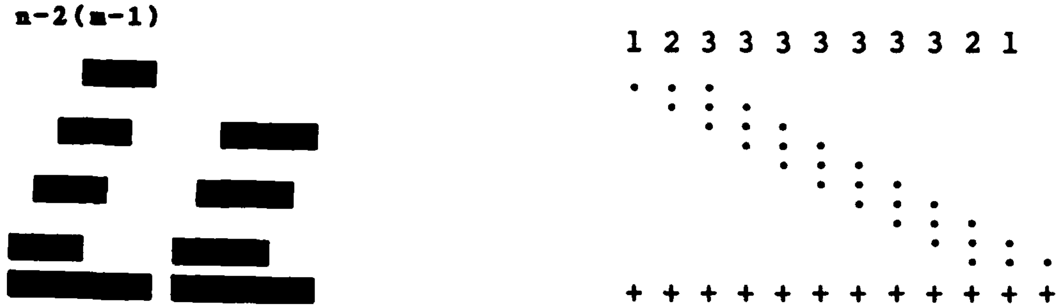
λ : $2[\ (1/\lambda) + (2/\lambda) + \ldots (m-1/\lambda)\ ] + \xi(m/\lambda) = 1$:

$\lambda = qp =$

$m^2 + m(\xi-1)$

ξ, the number of X_i that correspond to the maximum likelihood:

$\xi = n-2(m-1)$



[FIG. 1-A Fuzzy estimation]

We also see that order relations are preserved in a probabilistic sense:

where Y are the fuzzy estimators of X, and are of 'size' q,

$\gamma < q, n-q$

$P(X_1 < X_2 \mid Y_1 = \alpha,\ Y_2 = \alpha + \gamma) = (\omega^2 + \omega - \tau^2 - \tau + 2q\gamma)\ /\ 2q^2,$

$\omega = q-1, \quad \tau = \gamma - 1$

$P(X_1 = X_2 \mid Y_1 = \alpha,\ Y_2 = \alpha + \gamma) = q-\gamma\ /\ q^2$

$P(X_1 > X_2 \mid Y_1 = \alpha,\ Y_2 = \alpha + \gamma) = \sigma^2 + \sigma\ /\ 2q^2, \quad \sigma = q-\gamma-1$

2. A Generalized Rumelhart Net: GR-Net

Here we provide a functional outline of the Rumelhart network. Note that there is an explicit internal (hidden) layer, (the backward propogation).

$GR_i = \{ I, X, Z, c, s(t_0) \}$

I Input vector

X Intermediate (inner/hidden) level vector

Z Output vector

c is the comparison function (compares for identity or similarity)

$S(t_0)$ initial state By this we mean all the X(i),Z(i) are in some known, initial value (perhaps null)

Moreover, the X and Z can be further partitioned as follows:

$X = \{ S_x, I_x, V_x \}$

S_x, state of vectors corresponding to a specific node (likely unique for each node k, $(S_{11} \ldots S_{1n}) \ldots (S_{m1} \ldots S_{mn})$

I_x, state of inputs (copies) from previous level I: identical for each node k, $(I_{11} \ldots I_{1n}) \ldots (I_{m1} \ldots I_{mn})$

V_x, value of the vectors corresponding to each specific node a function of S_x and I_x, $(V_{11} \ldots V_{1n}) \ldots (V_{m1} \ldots V_{mn})$

n nodes previous level (I), m nodes current level (X)

where

$V_{ij} = f_1(S_{ij}, I_{ij})$

$X_q = f_2(V_{q1} \ldots V_{qn})$, $X = \{X_1 \ldots X_n\}$

$S_{ij}(t+1) = f_3(C_z, S_{ij}(t))$, some function of the the comparison function, and current state.

$Z = \{ R_z, X_z, W_z \}$

R_z, state of vectors corresponding to a specific node (likely unique for each node k, $(R_{11} \ldots R_{1m}) \ldots (R_{y1} \ldots R_{ym})$

X_z, state of inputs (copies) from previous level X: (identical for each node k), $(X_{11} \ldots X_{1m}) \ldots (X_{y1} \ldots X_{ym})$

W_z, value of the vectors corresponding to each specific node a function of R_z and V_z, $(W_{11} \ldots W_{1m}) \ldots (W_{y1} \ldots W_{ym})$

m nodes previous level (X), y nodes current level (Z)

where

$W_{ij} = f_4(R_{ij}, X_{ij})$

$Z_q = f_5(W_{q1} \ldots W_{qn})$

$Z = \{Z_1 \ldots Z_n\}$

$R_{ij}(t+1) = f_6(C_z, R_{ij}(t), X_{ij})$ or some similar function.

$C_z = f_7(Z_1 \ldots Z_y, T_1 \ldots T_y)$ the comparison function

The learning phase takes place in stages:

(I(i),T(i)) presented by the system

$X(i) = f_1(I(i))$

$Z(i) = f_2(X(i))$

$c(i) = f_3(Z(i),T(i))$

$Z(i') = f_4(c(i),Z(i))$

$X(i') = f_5(c(i),X(i))$

i refers to beginning state i' refers to ending state

T(i) is the target output value at time i paired with the input I(i) at time i. (Although in practice the system can be pipelined we show only one time period here for simplicity) temporally:

$V_{ij} = f_1(S_{ij}, I_{ij})$

$X_q = f_2(V_{q1} \ldots V_{qn}) \quad X = \{X_1 \ldots X_n\}$

$W_{ij} = f_4(R_{ij}, X_{ij})$

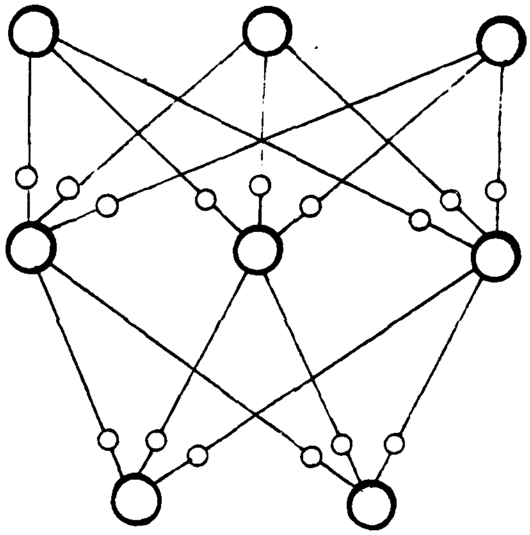
$Z_q = f_5(W_{q1} \ldots W_{qn}), \; Z = \{Z_1 \ldots Z_n\}$

$C_z = f_7(Z_1 \ldots Z_y, T_1 \ldots T_y)$ the comparison function

$R_{ij}(t+1) = f_6(C_z, R_{ij}(t), X_{ij})$ or some similar function

$S_{ij}(t+1) = f_3(C_z, S_{ij}(t), I_{ij})$ or some similar function

$T_1 \ldots T_y$ is the target output.



[FIG.A-1 A simple GR-Net: Large nodes correspond to I,X,Z (top to bottom), small nodes correspond to (V,S) and (W,R) (middle and bottom), the comparison function is not shown]

3. Statistical Aspects of Feature recognition

Consider a system with n images, (therefore $n^2 - n$ comparisons), and k possible features. Clearly, every image either has, or does not have a given feature. We encode each image as an incidence array $I_k = X_1 \ldots X_p$, where every $X_i \epsilon \{0,1\}$, with X_i equal to zero if feature i is not present and one if it is. Once all the images in the database have been coded in this way, we determine the corresponding probabilities $\pi_1 \ldots \pi_k$ of the features over the entire database. The probability that there are no matches, that is, the probability that we can uniquely determine every image in the database is approximately:

$$(1-z)^w, \; w = n^2 - n,$$

$$z = ((\pi_1)^2 + (1-\pi_1)^2)\;.\;.\;.((\pi_k)^2 + (1-\pi_k)^2))$$

Note that we can view a relation between two objects, or a set of relations as a feature, in the same way we view an object as a feature. In this way, we can generate as many features as we wish in order to obtain a unique determination of images.

4. Hamming model for templates

In its most basic form a Hamming distance simply represents the number of differences of two (n-bit) vectors, e.g.

$X = X_{1\ldots n}$, $Y = Y_{1\ldots n}$

$\Delta h(X_i, Y_j) = \Delta(X_i(1), Y_j(1)) + \;.\;.\;. + \Delta(X_i(n), Y_j(n))$

$\Delta(X(k), Y(h)) = 0$ if $X(k) = Y(h)$ else 1

So in a template matching context:

$\beta = X_i: \min_{ij}(\Delta_h(X_i, T))$, $\tau = T_j: \min_{ij}(\Delta_h(X_i, T))$,

$\xi = \Delta_h(\beta, \tau)$, $\Delta h(X_i, Y_j) = \Delta(X_i(1), Y_j(1)) + \;.\;.\;. + \Delta(X_i(n), Y_j(n))$

if $\xi > \sigma$, σ some (small) constant indicates excessive noise, the item seen is not found in the template library.

if $\xi > \sigma$ this is a no match condition.

if there is more than one β or τ this indicates a conflict (i.e. multiple matches), which is in effect also a no match condition.

END
DATE
FILMED
6-1988
DTIC